

## CLAIMS

What is claimed is:

1. A method for identifying one or more optimum configurations of a data processing system, said method comprising:
  - determining if a current configuration of said data processing system is optimized for system performance utilizing testing criteria;
  - in response to determining said current configuration is not optimized, generating alternate configurations;
  - analyzing said alternate configurations utilizing said testing criteria to identify at least one configuration optimized for system performance;
  - in response to identifying at least one configuration optimized for system performance, presenting said at least one configuration optimized for system performance to a user; and
  - in response to not identifying a configuration optimized for system performance, altering said testing criteria and again analyzing said alternate configurations.
2. The method according to claim 1, wherein said data processing system includes a plurality of hardware adapters coupled to an interconnect, and wherein generating alternate configurations comprises:
  - generating alternate connections between said plurality of hardware adapters and said interconnect.
3. The method according to claim 1, said generating further includes:
  - constructing a tree structure in which said current configuration is a root node and said alternate configurations are child nodes of said root node.

1           4.     The method according to claim 3, wherein said testing criteria comprises first  
2 testing criteria, said method further comprising:

3                 rejecting an alternate configuration if said alternate configuration does not meets  
4 said first testing criteria.

1           5.     The method according to claim 4, wherein said rejecting further comprises:  
2                 testing said alternate configuration for improper placement of at least a plurality  
3 of hardware adapters on a plurality of interconnect segments.

1           6.     The method according to claim 5, wherein said testing further includes:  
2                 for a candidate configuration, designating a fastest setting supported by said  
3 interconnect as a chosen setting;

4                 determining whether or not a number of a plurality of hardware adapters that  
5 support said chosen setting (NADAPTERS) equals a number of a plurality of  
6 interconnect segments coupled to at least a hardware adapter that supports said chosen  
7 setting (NSEGMENTS);

8                 in response to determining NADAPTERS does not equal NSEGMENTS,  
9 determining whether or not at least one of said plurality of hardware adapters can be  
10 moved to an empty one of said plurality of interconnect segments; and

11                in response to determining at least one of said plurality of hardware adapters can  
12 be moved to an empty one of said plurality of interconnect segments, determining said  
13 candidate configuration is not an optimal configuration.

1           7.     The method according to claim 6, wherein said testing further comprises:  
2                 in response to determining NADAPTERS equals NSEGMENTS, determining  
3 whether or not there is at least a slower setting than said chosen setting;

4                 in response to determining there is at least a slower setting, designating said  
5 slower setting as said chosen setting;

6 in response to determining there is not at least a slower setting, determining said  
7 candidate configuration is said optimal configuration.

1 8. The method according to claim 3, said constructing further includes:

2 constructing said tree structure such that said alternate configurations are  
3 variations of said current configuration obtained by performing one alteration to said  
4 current configuration.

1 9. A system for identifying one or more optimum configurations of a data  
2 processing system, said system comprising:

3 an interconnect;

4 a processor, coupled to said interconnect;

5 a memory coupled to said processor; and

6 a system resource optimizer resident in said memory and executable by said  
7 processor to determine if a current configuration of said data processing system is  
8 optimized for system performance utilizing testing criteria, wherein responsive to a  
9 determination that said current configuration is not optimized, said system optimizer  
10 generates alternate configurations and analyzes said alternate configurations with said  
11 testing criteria to identify at least one configuration optimized for system performance  
12 that is presented to a user, and wherein responsive to failing to identify at least one  
13 configuration optimized for system performance, said system optimizer alters said testing  
14 criteria and again analyzes said alternate configurations.

1 10. The system according to claim 9, wherein said data processing system includes  
2 a plurality of hardware adapters coupled to an interconnect, and said system optimizer  
3 comprises:

4 means for generating alternate connections between said plurality of hardware  
5 adapters and said interconnect.

1 11. The system according to claim 9, said system optimizer further comprising:  
2 means for constructing a tree structure in which said current configuration is a  
3 root node and said alternate configurations are child nodes of said root node.

1 12. The system according to claim 11, wherein said testing criteria comprises first  
2 testing criteria, said system further comprising:  
3 means for rejecting an alternate configuration if said alternate configuration does  
4 not meet said first testing criteria.

1 13. The system according to claim 12, wherein said means for rejecting further  
2 comprises:  
3 means for testing said alternate configuration for improper placement of at least  
4 a plurality of hardware adapters on a plurality of interconnect segments.

1 14. The system according to claim 13, wherein said means for testing further  
2 includes:

3 for a candidate configuration, means for designating a fastest setting supported  
4 by said interconnect as a chosen setting;

5 means for determining whether or not a number of a plurality of hardware  
6 adapters that support said chosen setting (NADAPTERS) equals a number of a plurality  
7 of interconnect segments coupled to at least a hardware adapter that supports said chosen  
8 setting (NSEGMENTS);

9 means, responsive to determining NADAPTERS does not equal NSEGMENTS,  
10 for determining whether or not at least one of said plurality of hardware adapters can be  
11 moved to an empty one of said plurality of interconnect segments; and

12 means, responsive to determining at least one of said plurality of hardware  
13 adapters can be moved to an empty one of said plurality of interconnect segments, for

14 determining said candidate configuration is not an optimal configuration.

1 15. The system according to claim 14, wherein said means for testing further  
2 comprises:

3 means for determining whether or not there is at least a slower setting than said  
4 chosen setting, if NADAPTERS equals NSEGMENTS;

5 means, responsive to determining there is at least a slower setting, for designating  
6 said slower setting as said chosen setting;

7 means, responsive to determining there is not at least a slower setting, for  
8 determining said candidate configuration is said optimal configuration.

1 16. The system according to claim 11, said means for constructing further  
2 comprising:

3 means for constructing said tree structure such that said alternate configurations  
4 are variations of said current configuration obtained by performing one alteration to said  
5 current configuration.

1 17. A computer program product comprising:  
2 a computer-usable medium; and  
3 a system resource optimizer encoded within said computer-usable medium to  
4 determine if a current configuration of said data processing system is optimized for  
5 system performance utilizing testing criteria, wherein responsive to a determination that  
6 said current configuration is not optimized, said system optimizer generates alternate  
7 configurations and analyzes said alternate configurations with said testing criteria to  
8 identify at least one configuration optimized for system performance that is presented to  
9 a user; and wherein responsive to failing to identify at least one configuration optimized  
10 for system performance, said system optimizer alters said testing criteria and again  
11 analyzes said alternate configurations.

12 18. The computer program product according to claim 17, wherein said data  
13 processing system includes a plurality of hardware adapters coupled to an interconnect,  
14 and wherein said system optimizer further comprises:

15 instructions, encoded within said computer-usable medium, for generating  
16 alternate connections between said plurality of hardware adapters and said interconnect.

17 19. The computer program product according to claim 17, said instructions for  
18 generating further includes:

19 instructions, encoded within said computer-usable medium, for constructing a  
20 tree structure in which said current configuration is a root node and said alternate  
21 configurations are child nodes of said root node.

22 20. The computer program product according to claim 19, wherein said testing  
23 criteria comprises first testing criteria, said computer program product further comprises:

24 instructions, encoded within said computer-usable medium, for rejecting an  
alternate configuration if said alternate configuration does not meet said first testing

5 criteria.

1 21. The computer program product according to claim 20, wherein said instructions  
2 for rejecting further comprises:

3 instructions, encoded within said computer-usable medium, for testing said  
4 alternate configuration for improper placement of at least a plurality of hardware adapters  
5 on a plurality of interconnect segments.

1 22. The computer program product according to claim 21, wherein said instructions  
2 for testing further includes:

3 for a candidate configuration, instructions, encoded within said computer-usable  
4 medium, for designating a fastest setting supported by said interconnect as a chosen  
5 setting;

6 instructions, encoded within said computer-usable medium, for determining  
7 whether or not a number of a plurality of hardware adapters that support said chosen  
8 setting (NADAPTERS) equals a number of a plurality of interconnect segments coupled  
9 to at least a hardware adapter that supports said chosen setting (NSEGMENTS);

10 in response to determining NADAPTERS does not equal NSEGMENTS,  
11 instructions, encoded within said computer-usable medium, for determining whether or  
12 not at least one of said plurality of hardware adapters can be moved to an empty one of  
13 said plurality of interconnect segments; and

14 in response to determining at least one of said plurality of hardware adapters can  
15 be moved to an empty one of said plurality of interconnect segments, instructions,  
16 encoded within said computer-usable medium, for determining said candidate  
17 configuration is not an optimal configuration.

1 23. The computer program product according to claim 22, wherein said instructions  
2 for testing further comprises:

3           in response to determining NADAPTERS equals NSEGMENTS, instructions,  
4 encoded within said computer-usable medium, for determining whether or not there is  
5 at least a slower setting than said chosen setting;

6           in response to determining there is at least a slower setting, instructions, encoded  
7 within said computer-usable medium, for designating said slower setting as said chosen  
8 setting;

9           in response to determining there is not at least a slower setting, instructions,  
10 encoded within said computer-usable medium, for determining said candidate  
11 configuration is said optimal configuration.

24.    The computer program product according to claim 19, said constructing further  
includes:

instructions, encoded within said computer-usable medium, for constructing said  
tree structure such that said alternate configurations are variations of said current  
configuration obtained by performing one alteration to said current configuration.